Numerische Methoden der Festkörperphysik
Übung 1, Lösungsvorschlag

Nils Blümer, Eberhard Jakobi

AG KOMET337
Universität Mainz

9. Mai 2007

Überblick

Zufallszahlen

Einfache Zufallszahlen

```
int rand ( void );
```

Returns a pseudo-random integral number in the range 0 to RAND_MAX. This number is generated by an algorithm that returns a sequence of apparently non-related numbers each time it is called.
Usage for creation of doubles

```
double r = rand()/((double)(RAND_MAX));
```

`(double)` stellt sicher, dass eine "double"-Division durchgeführt wird! Solch eine Operation bezeichnet man als `cast` (eingedeutscht casten).

Metropolis-Programm

Realisierung der Potentialfunktion

```
double potential(double x)
{
   if (x<-3. || x>3.) {
      return std::numeric_limits<double>::max();
   } else {
      return
      -(std::pow(x,2)+.3)*std::exp(-std::pow((x-.1),2));
   }
}
```

Metropolis-Programm

Implementierung der Metropolisbedingung

```
bool inline
metropolis(double beta, double E_n, double E_c)
{
   if (E_n<E_c) return true;
   double r = (double)(rand())/((double)(RAND_MAX));
   if (std::exp(-beta*(E_n-E_c))>r)
       return true;
   else
       return false;
}
```

Metropolis-Programm

Iterationszyklus

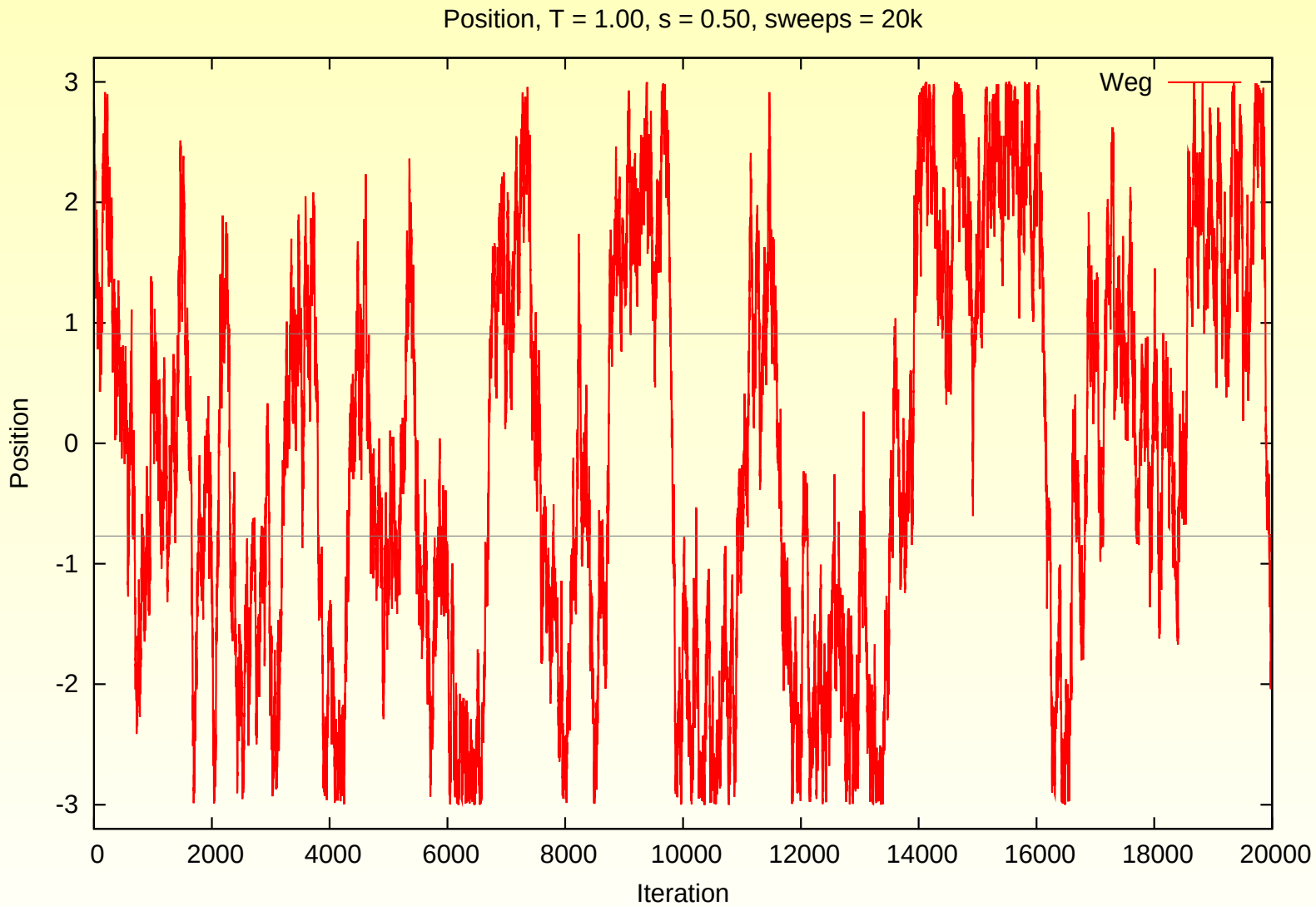
```
for (size_t iter = 0; iter < max_iter; iter++) {
   double r = (double)(rand())/((double)(RAND_MAX));

   double x_neu = x_akt + (r-.5)*s;
   double E_neu = potential(x_neu);

   if (metropolis(beta,E_neu,E_akt)) {
      x_akt = x_neu;
      E_akt = E_neu;
      accepted++;
   }

   x_array[iter] = x_akt;
   E_array[iter] = E_akt;
}
```

Metropolis-Programm

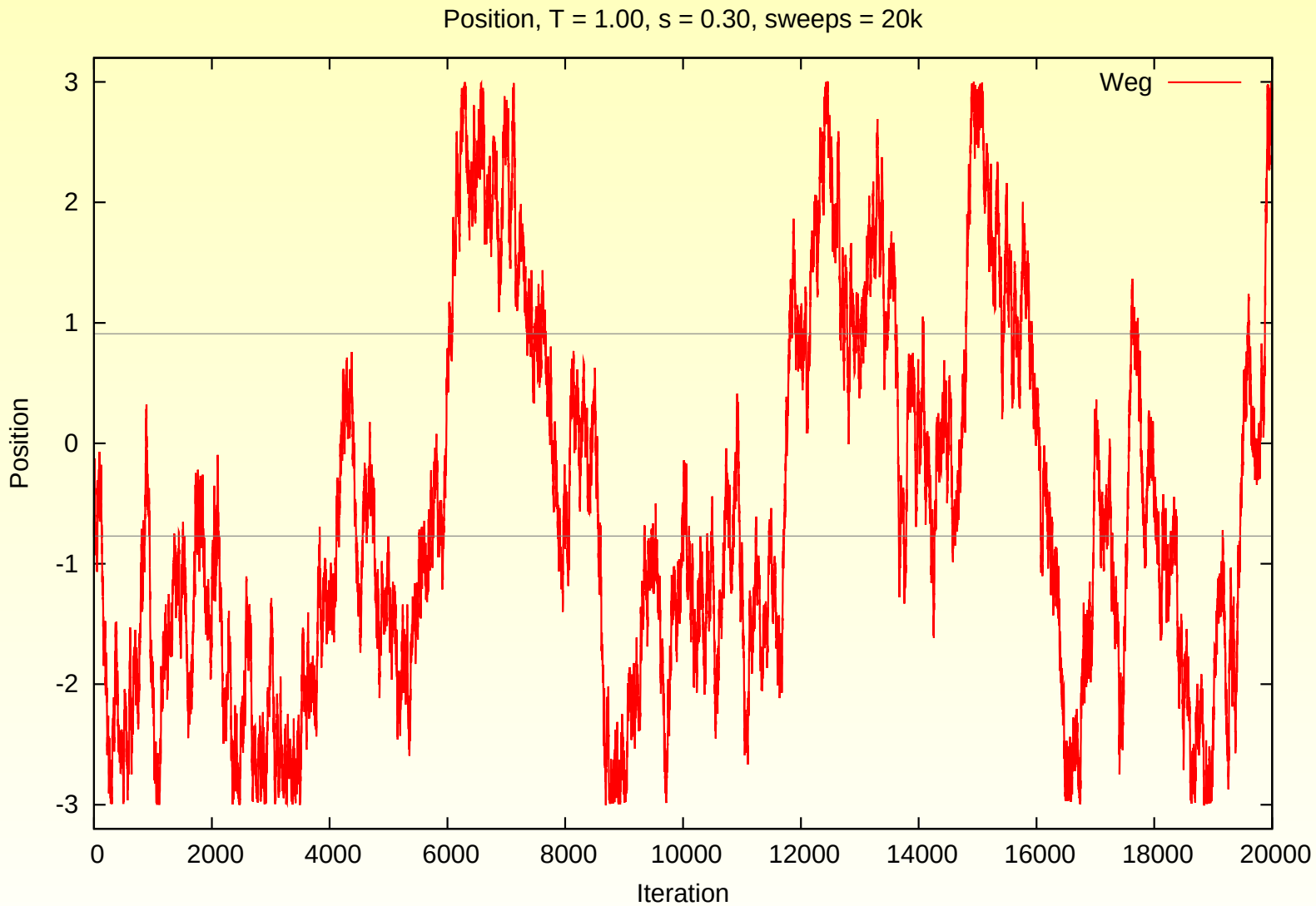
Ausgabe der Daten mit Hilfe von C++ Streams

```
std::ofstream x_stream("X");
std::ofstream E_stream("E");
for(size_t i = 0; i < max_iter; i++) {
   x_stream << x_array[i] << "\n";
   E_stream << E_array[i] << "\n";
}
x_stream.close();
E_stream.close();
```

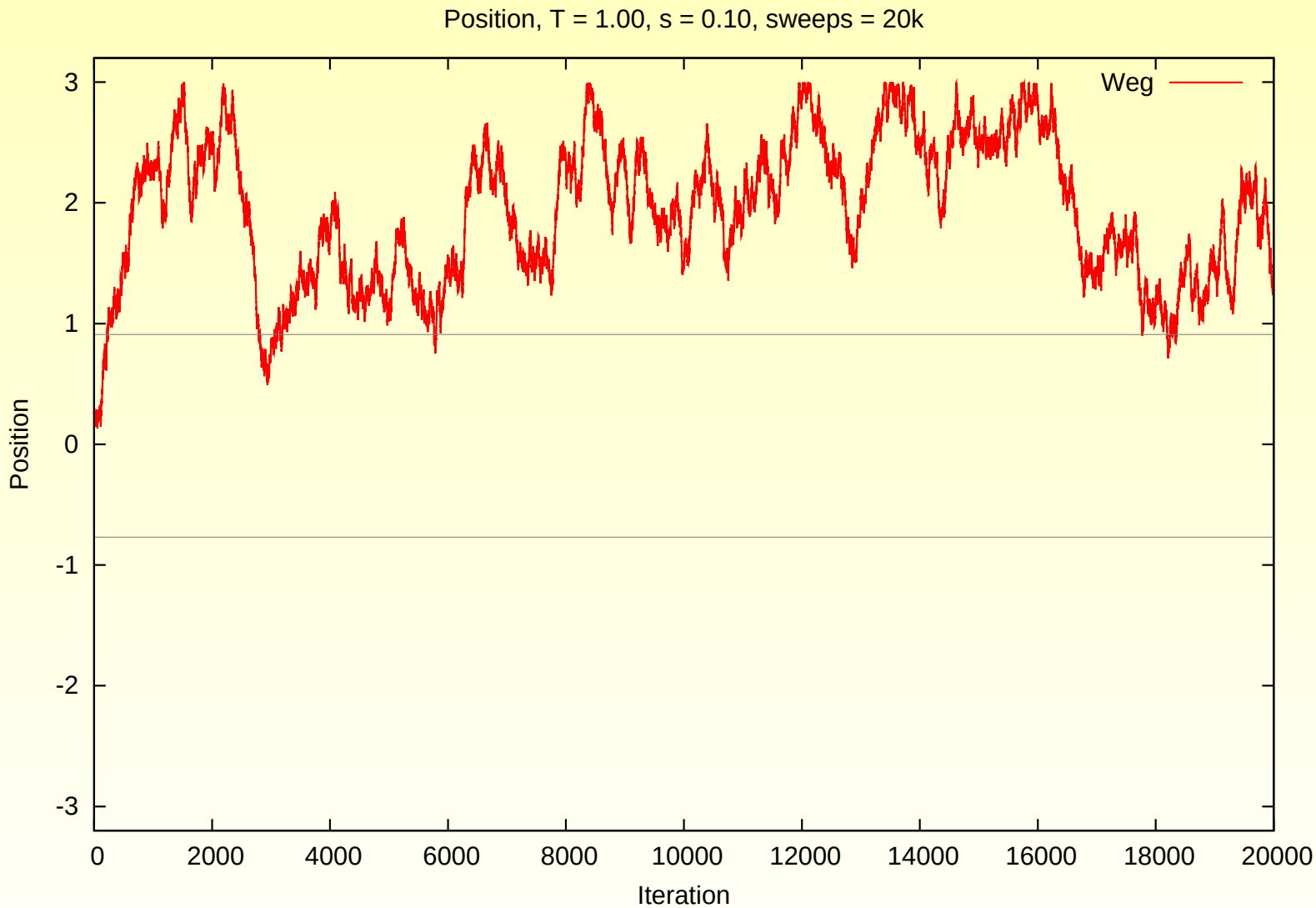
Einfluss der Schrittlänge



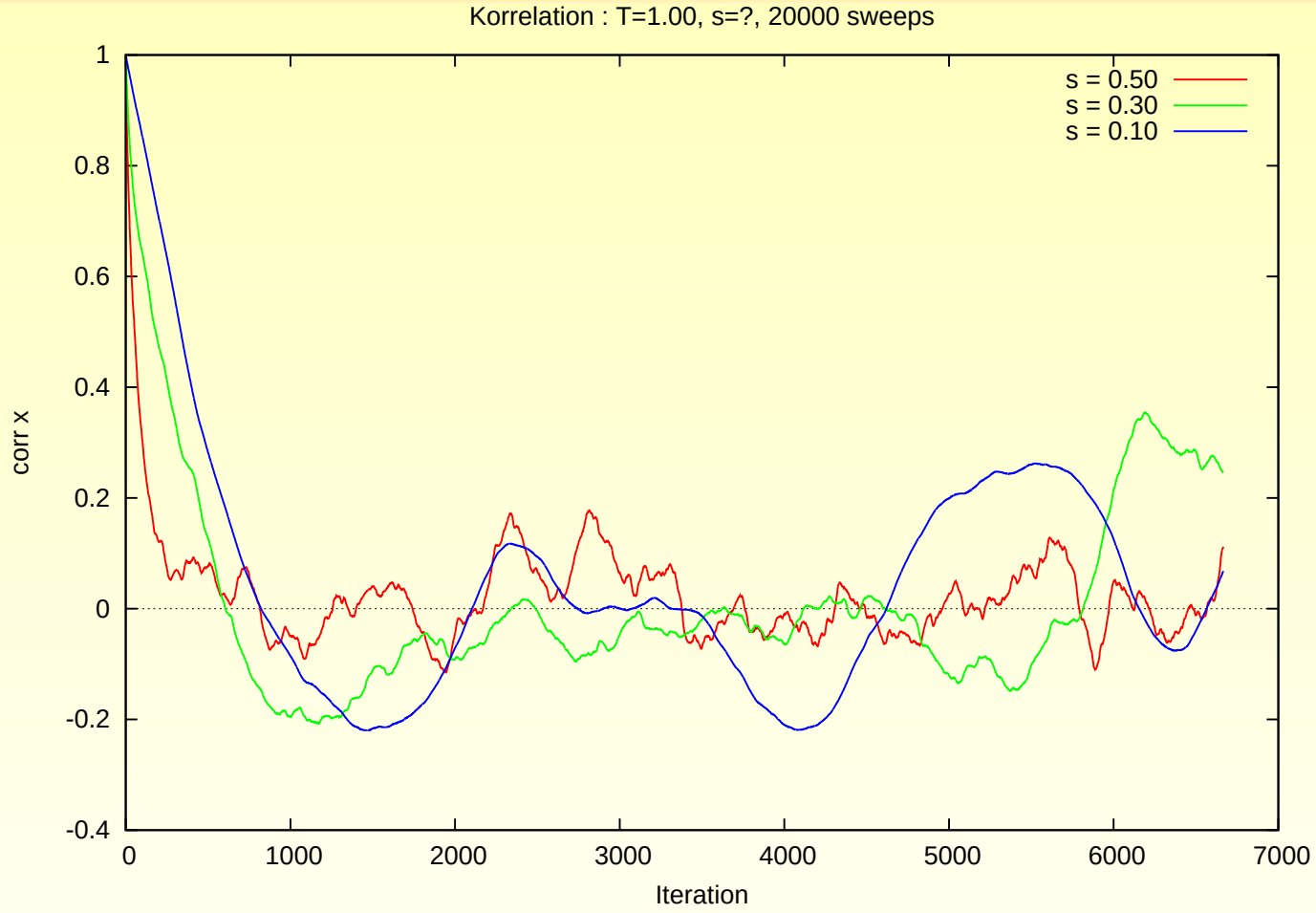
Einfluss der Schrittlänge



Einfluss der Schrittlänge

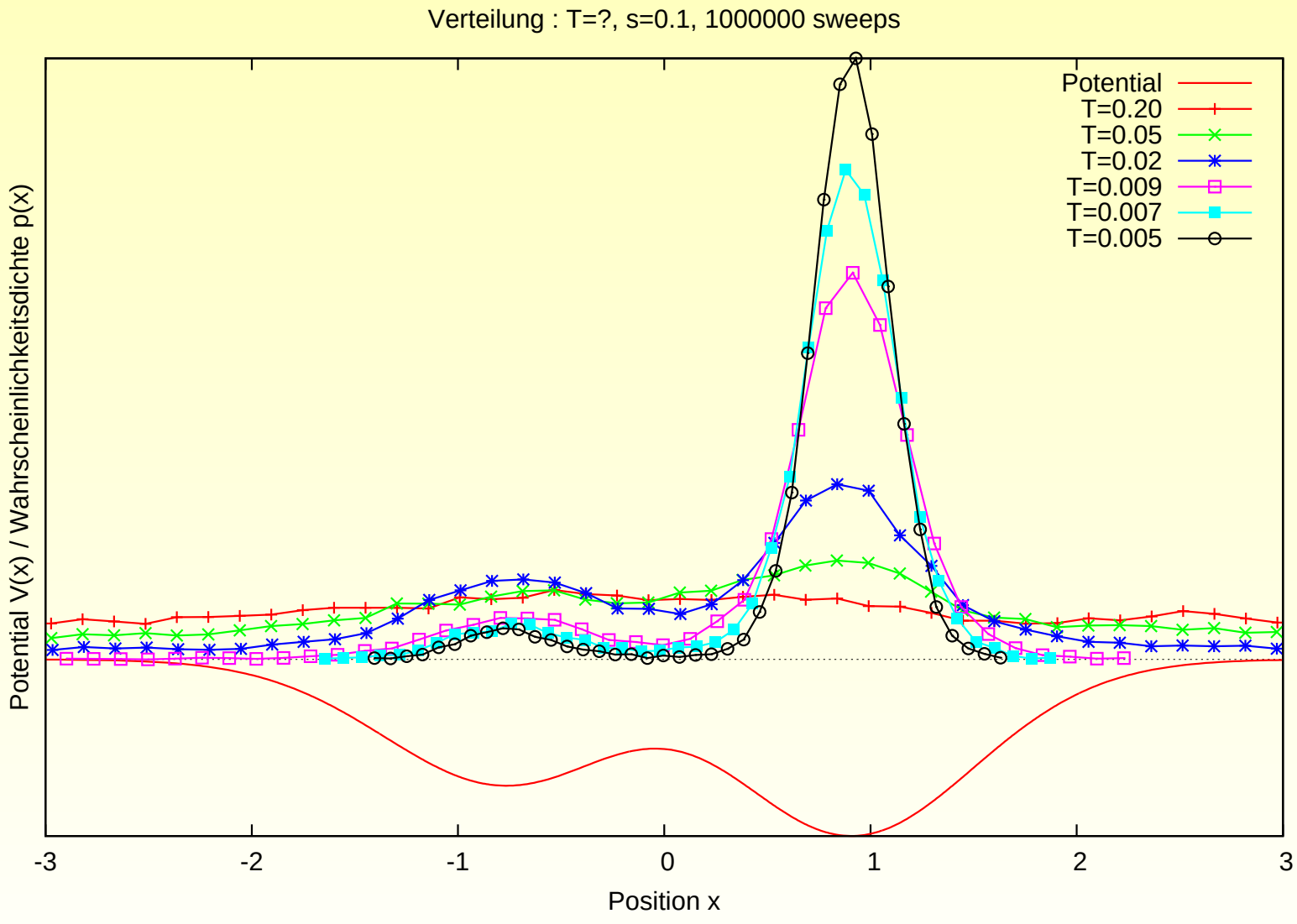


Einfluss der Schrittlänge auf die Korrelationen

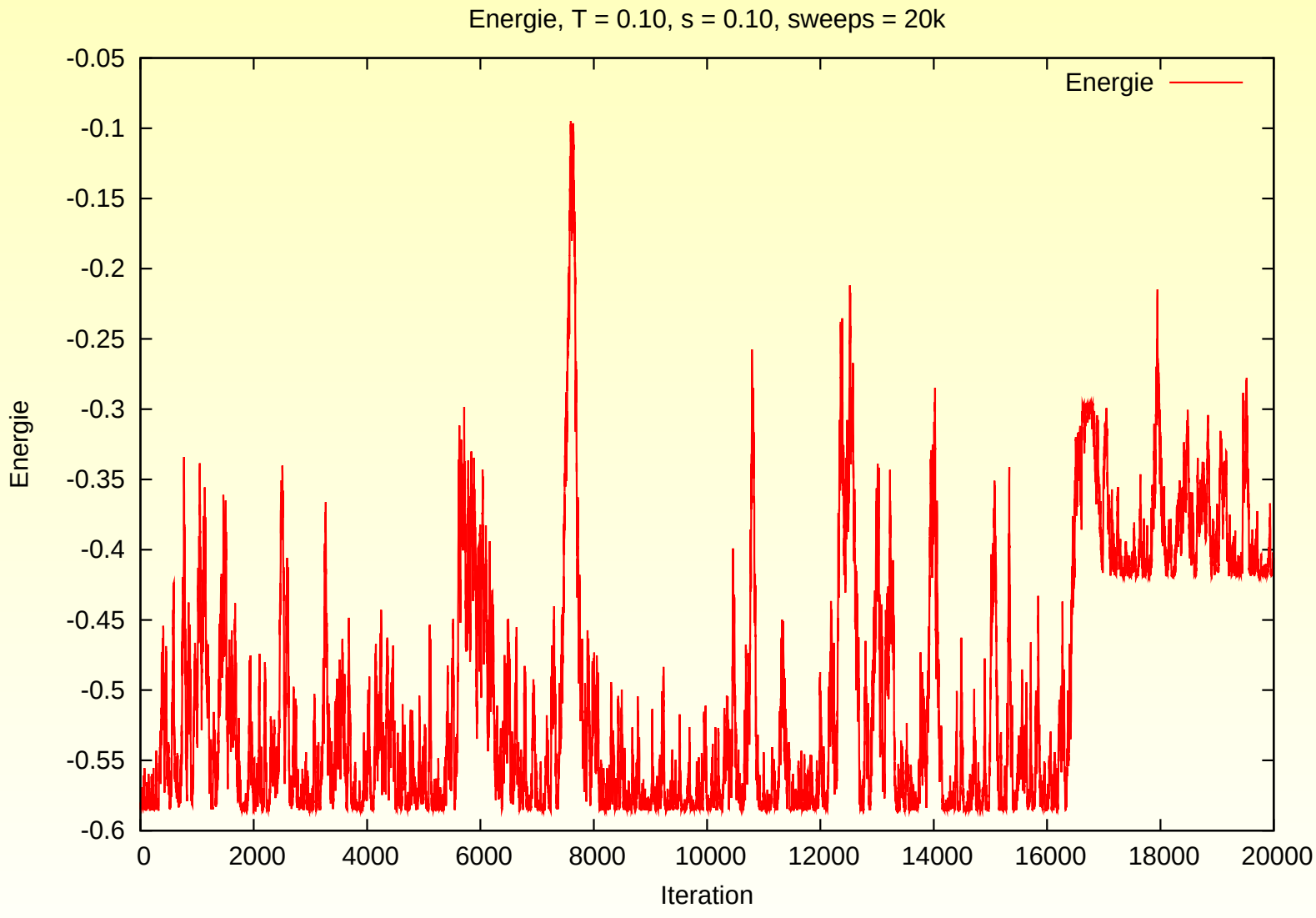


Autokorrelationszeit $\tau(s = 0.5) = 688$
Autokorrelationszeit $\tau(s = 0.3) = 1291$
Autokorrelationszeit $\tau(s = 0.1) = 2745$

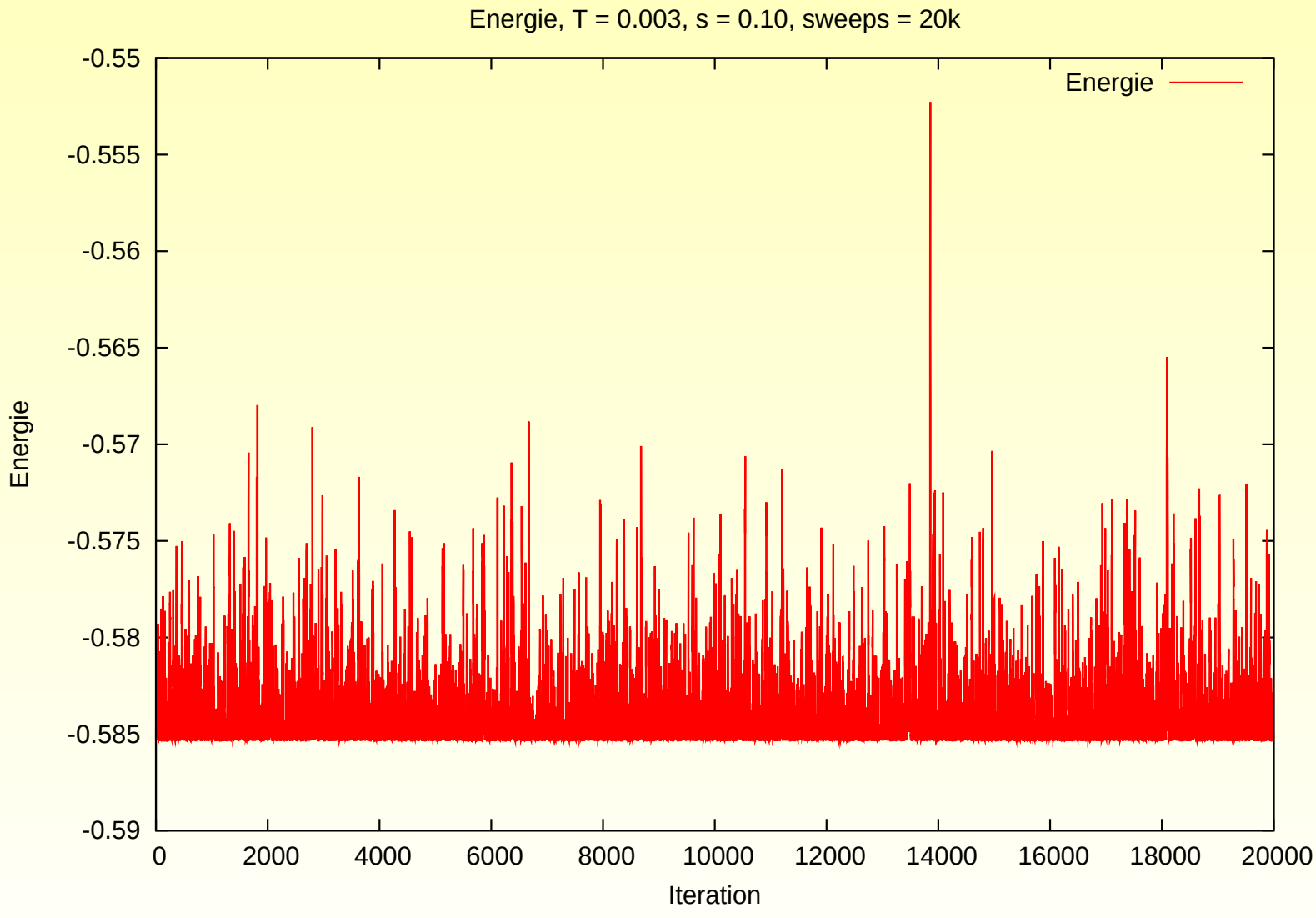
Einfluss der Temperatur



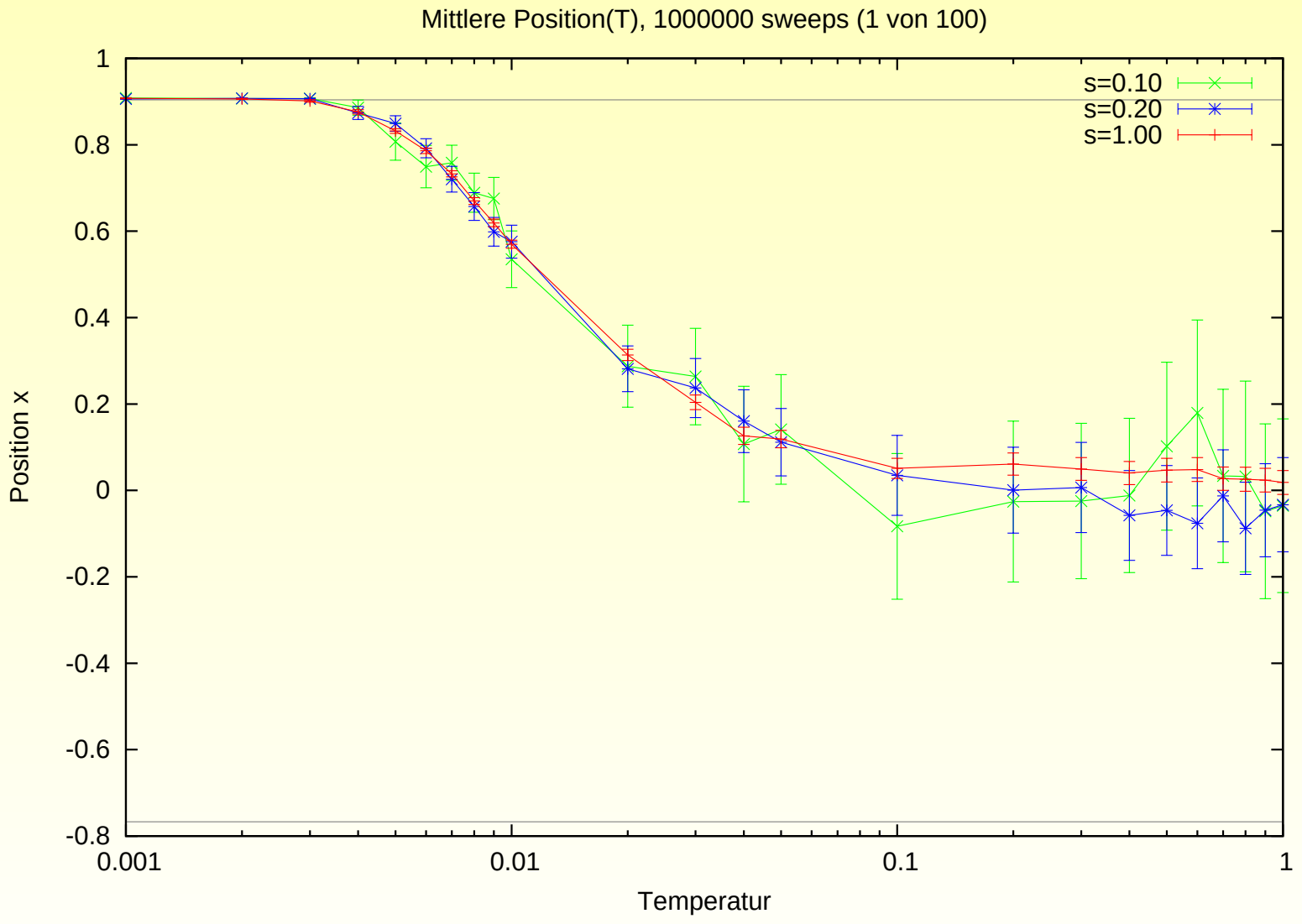
Energie für höhere Temperatur



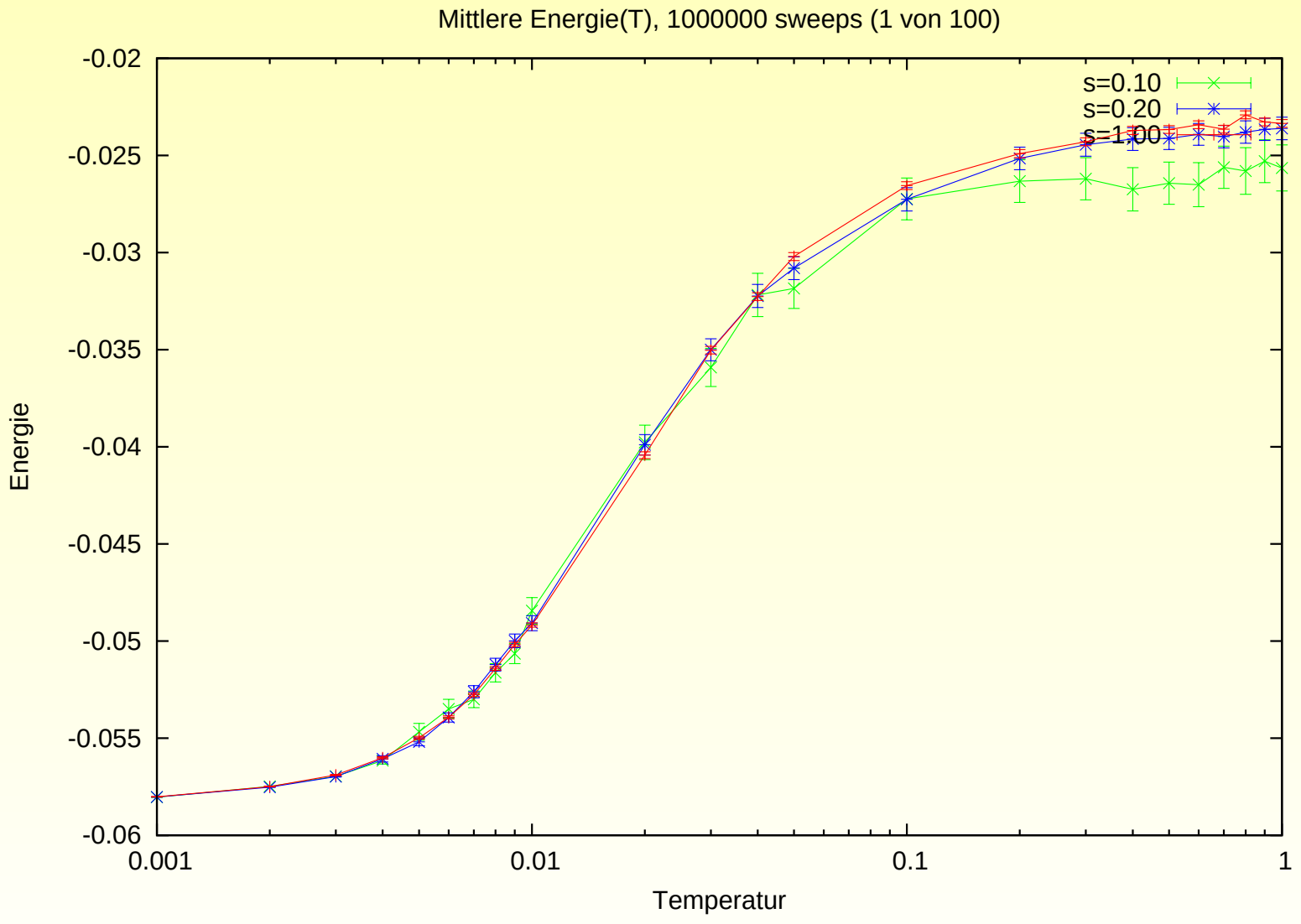
Energie für niedrige Temperatur



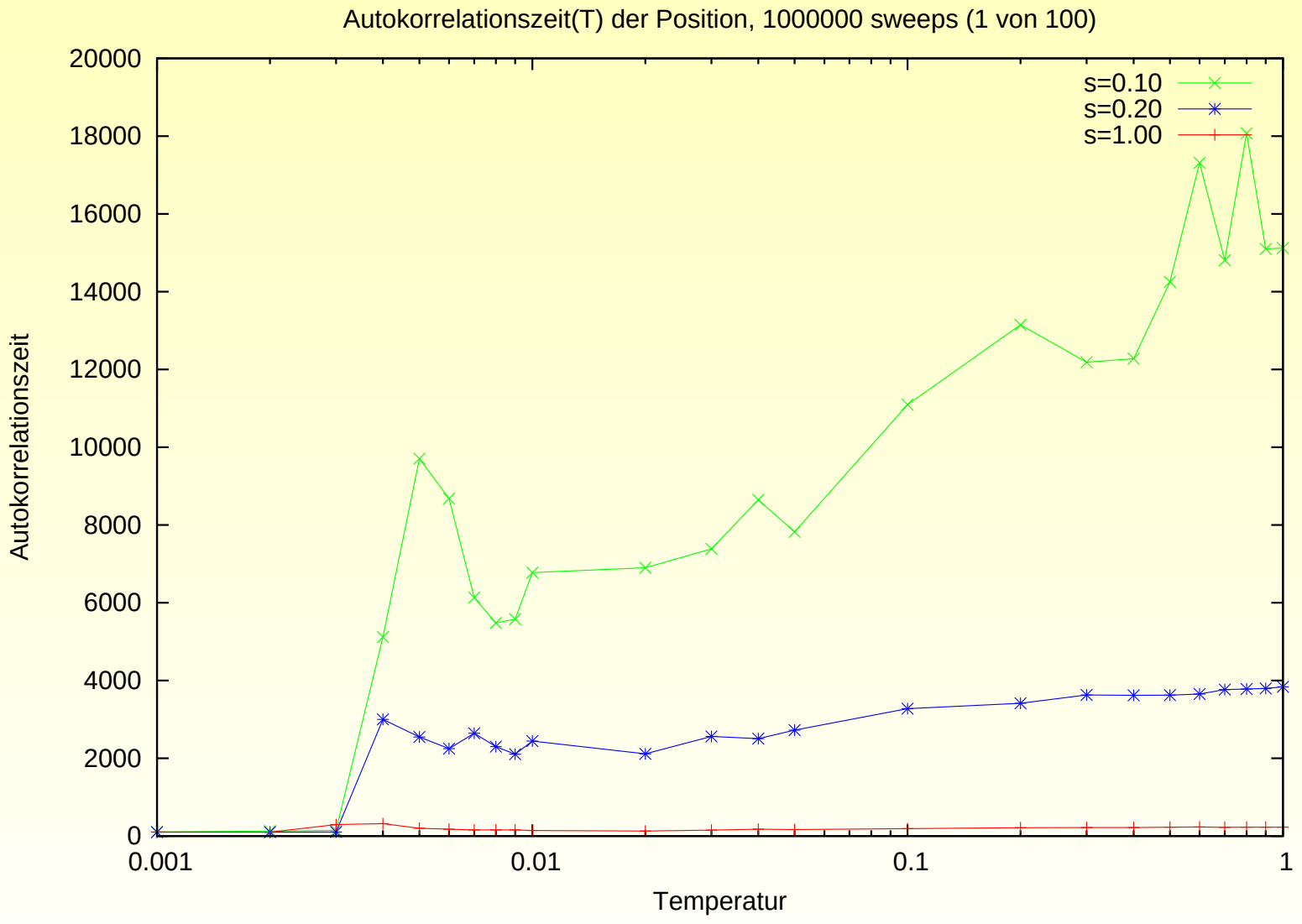
Temperaturabhängigkeit der Größen, Mittlere Position



Temperaturabhängigkeit der Größen, Energie



Autokorrelationszeit der Position(T)



Autokorrelationszeit der Energie(T)